PHASE CORRELATION BASED MOTION ESTIMATION IN HYBRID VIDEO COMPRESSION

INVENTORS: Deepika Srinivasan, Andrew Krupiczka, and Sassan Pejhan

FIELD OF INVENTION

[0001] This invention relates in general to video compression, and more particularly to motion estimation in video images.

BACKGROUND OF INVENTION

[0002] Motion Estimation (ME) and Compensation is an important technique to exploit the temporal correlations among successive frames in a video sequence. Almost all current video compression standards such as MPEG-1/2/4 and H.26x employ a hybrid of block-based motion compensated prediction and transform coding for representing variations in picture content due to moving objects. In block-based motion estimation, a current frame is divided into rectangular blocks and an attempt is made to match each current block with a block from a reference frame, which would serve as the predictor of the current block. The difference between this predictor block and the current block is then encoded and transmitted. The (x,y) offset of the current block from the predictor block is characterized as a motion vector. A significant improvement in compression efficiency is achieved since usually the 'difference block' has a much lower energy or information content than the original block.

[0003] The improvement in compression efficiency, however, comes at a significant increase in complexity, since the process of matching a current block with a predictor block almost always involves a search algorithm. The current block is searched for the best possible match in the reference frame within a search window located around the position of the block in the current frame. For each search location, some metric – typically the Sum of Absolute Differences (SAD), or the Sum of Squared Difference (SSD) between the pixels of the two blocks – is calculated. The block that produces the smallest value in the metric is then selected as the predictor block. A full search strategy typically involves testing all the available blocks in the search range leading to a high computational complexity. The complexity of the search algorithm thus depends on the size of the search area (amongst other things).

[0004] The algorithms aimed at simplifying the number of calculations for motion estimation can be classified as being pel-recursive, block-based or object based. The pel-recursive methods lead to a significant number of operations per frame, as calculations have to be done on every pixel. The object-based methods involve separate operations for object-recognition leading to computational complexity. It has been observed that the computational complexity could be reduced if efficient block-based search techniques could be designed.

[0005] Many attempts aimed at reducing the complexity of ME have focused on Fast Motion Estimation (FME) algorithms, which focus on ways to reduce the number of search candidates required to find a 'good match' while leading to a minimum degradation in the predicted video quality as compared to the exhaustive search. Several block-based motion estimation algorithms that are computationally faster than the full search have been investigated and developed. The three-step search (TSS), new three-step search (NTSS), four step-search (4SS), block-based gradient descent search (BBGDS), diamond search (DS), hexagon-based search (HEXBS), and Unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) are a few such FME algorithms. In addition, various FME methods are also disclosed in U.S. Patent Nos. 6668020, 6542547, 6414997, 6363117, 6269174, 6259737, 6128047, 5778190, 5706059, and 5557341. In general, these methods are carried out in the spatial domain and depend on the shape and size of the search pattern and on the efficient choice of the search center to increase the speed of the motion vector search. However, the disadvantage is that these techniques may fall into a local distortion minimum and not identify the best predictor block. Also, the reduction in the number of search points depends on the shape of the search pattern.

[0006] While FME algorithms can reduce the complexity of the ME process by a factor of 10 or better, they nonetheless suffer from the fact that – like the full search algorithm – their complexity is proportional to the size of the search area. For a highly complex encoder – such as the latest H.264/MPEG-4 AVC encoder – a factor of 10 or 20 improvement may still not be sufficient for real-time performance. There is therefore a need for an alternative mechanism that can perform motion estimation at much lower complexity, but without sacrificing compression efficiency.

SUMMARY OF THE INVENTION

[0007] A phase correlation based analysis is used to identify a number of candidate motion vectors for predicting macroblocks with a predicted frame. For a macroblock that is to be predicted, the candidate motion vectors are evaluated to determine which candidate results in the least distortion with respect to a reference frame. The phase correlation analysis is generally independent of the size of predicted frame or region thereof being analyzed, and thus readily scales to various sized images, without significantly increasing the time needed to encode the frames.

[0008] In one embodiment, a predicted frame and a reference frame are divided into phase correlation blocks, each of which includes a number of macroblocks. The size of the phase correlation blocks can be varied to tradeoff encoding speed and image quality, as well as to take advantage of the computational implementation of the phase analysis. A phase analysis between corresponding phase correlation blocks in the predicted frame and reference frame results in a phase correlation surface. The surface will have a number of peaks, each of which represents underlying motion between the two phase correlation blocks. A number of the most significant phase correlation peaks are selected. The motion vectors associated with these peaks are likewise determined, according to the location of the peaks on the phase correlation surface. To encode a macroblock within a phase correlation block, the candidate motion vectors are evaluated with respect to the macroblock to determine which candidate minimizes a distortion measure for the macroblock. This candidate is then selected for encoding the macroblock.

[0009] There are a number of various refinements of the above process, which can be used independently or in any combination. First, the phase correlation blocks can either be overlapping or non-overlapping. Second, a windowing function may be applied to the edges of the phase correlation blocks to smooth their pixel values, and reduce discontinuities between the blocks. Third, the number of candidate motion vectors can be fixed or variable. Where the number is variable, it can be selected based on the size of the frame, a variance of the phase correlation peaks, or other context dependent variables. Fourth, the evaluation of candidate motion vectors can be extended once a best candidate is selected to include an evaluation of subpixel nearby motion vectors. Fifth, the evaluation of candidate motion vectors can terminate before all of the candidates are evaluated, for example if a threshold distortion measure is reached; the threshold can be fixed or adaptive.

[0010] The present invention has various embodiments, including as a computer implemented process, as computer apparatuses, as integrated circuits, and as computer program products that execute on general or special purpose processors.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 is a block diagram of a hybrid encoder.

[0012] FIG. 2 is a flowchart of the motion estimation and motion compensation process.

[0013] FIG. 3 illustrates a frame with non-overlapping phase correlation blocks.

[0014] FIG. 4 illustrates a frame with overlapping phase correlation blocks.

[0015] FIG. 5 illustrates motion vector candidates on a phase correlation surface.

[0016] FIG. 6 illustrates an example of an extended 1D cosine bell window.

[0017] The figures depict various embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the illustrated and described structures, methods, and functions may be employed without departing from the principles of the invention.

DETAILED DESCRIPTION

[0018] Referring now to FIG. 1, there is shown a block diagram of an encoder 100 according to one embodiment of the present invention. The encoder 100 generally operates according to principles of an MPEG-1 or MPEG-2 video encoder as described in ISO/IEC 11172 and ISO/IEC 13818 or similar hybrid Motion Compensation/DCT encoder, with the additional features and structures as further described below. The encoder 100 is used to encode a video sequence that comprises a plurality of uncompressed images 102. The image 102 data comprises a plurality of macroblocks, each macroblock having either 4:x:x sample format of luminance and chrominance data, depending on the implementation.

[0019] The type of image (equivalently "frame" or "picture" or "slice") will be one of an intra-picture (I), a forward predicted picture (P), or a bi-directional predicted (B) picture. Intra-pictures (I-pictures) are coded without reference to other pictures contained in the video sequence. In older video compression standards such as MPEG-1 and MPEG-2, Inter-frame predicted pictures (P-pictures) are coded with reference to the nearest previously coded I-picture or P-picture, usually incorporating motion compensation to increase coding efficiency. In H.264/MPEG-4 AVC, different blocks within a P-slice may refer to blocks in different, previously coded I-slice or P-slice. Bidirectionally predicted (B-pictures) may use both past and future frames as references. To achieve high compression, motion compensation can be employed based on the nearest past and future P-pictures or I-pictures.

[0020] The encoder 100 uses different encoding processes depending on the type of the image. P and B type frames are encoded as follows. The image macroblocks are input into both subtractor 104 and motion estimator 122. The motion estimator 122 determines for each macroblock a motion vector, using the phase correlation analysis as described below, where the motion vector represents the horizontal and vertical displacement from the current macroblock being encoded to the matching macroblock in the reference picture. When completed for all of the macroblocks in the current image, the result is a set of motion vectors corresponding to the macroblocks of the current image. Each macroblock will be also coded as either a forward predicted (P), backward predicted (B), weighted average predicted (W), intra (I), or skip (S) block, depending on the motion vector information for the block, if any, and the frame type. For a B frame, motion predicted blocks will be either P or B blocks, depending on the reference picture used for the particular macroblock; otherwise, the blocks will be I or S blocks. For a P frame, the blocks will be either P, S, or I blocks. These motion vectors data and block information are passed to the motion compensation stage 124, which applies them to the reference image(s) to create a motion compensated image 126.

[0021] The motion compensated image 126 is then subtracted from the original image 102 by subtractor 104 to produce a set of error prediction or residual signals for each macroblock (in practice this step takes place on macroblock by macroblock basis directly following motion estimation). This error prediction signal represents the difference between the predicted image and the original image 102 being encoded. In the case that the original image 102 is a B- or P-picture, the motion compensated image 126 is an estimation of the original image 102. In the case that the original image 102 is an I-picture, then the motion compensated image 126 will have all pixel values being equal to zero, and the quantized DCT coefficients represent transformed either pixel values or error prediction signal between the current block and a previously encoded block in the current frame. For I frames, the quantized coefficients are used to reconstruct the I frame as a reference image via inverse quantizer 100 and inverse DCT 118. The reconstructed image is stored in frame buffer 120.

[0022] The motion estimator 122 calculates the motion vector of each block using a phase correlation method that directly measures the motion of image content. Generally, the method uses phase correlation between blocks of the current frame and the reference frame to identify a limited number of the best motion vectors for predicting the macroblocks of the current frame. This limited number of candidate motion vectors is then evaluated with respect to each macroblock to be predicted to determine the best motion vector. A particular benefit of this method is that computational complexity is independent of the size of the search area, thereby allowing for scaling and coding quality to be variably controlled.

[0023] Referring now to FIG. 2, there is shown a flowchart of the operation of the motion estimator 122 to perform the phase correlation-based motion estimation. The input to the motion estimator 122 is the uncompressed image 102 received from frame buffer 120. The motion estimator 122 divides 200 the reference frame R and the current frame P (equivalently, predicted frame) for which motion is to be estimated are divided into phase correlation blocks of size MxN, where M and N are a power of 2 in order to take advantage of existing FFT algorithms. Some hybrid encoders (e.g. those based on the H.263, MPEG-4 and H.264 standards) allow motion vectors to point outside of the boundaries of the frame (so called Unrestricted Motion Vector mode) to cater for objects that have partially moved outside of the frame. In such cases, the frames are extended on each border (through edge padding methods prescribed by the underlying standards) by the amount of the maximum motion vector allowed. If this option is enabled, then it is the 'extended frames' that are split into phase correlation blocks of size MxN. FIG. 3 illustrates an embodiment where the phase correlation blocks 300 are non-overlapping; each phase correlation block covers a number of macroblocks 302. FIG. 4 illustrates an embodiment where the phase correlation blocks 300 overlap, as further described below.

[0024] In one embodiment, the phase correlation block dimensions (M and N) are chosen to be the power of two greater than $2S+16$, where S is the maximum magnitude of the motion vectors, and describes a maximum search range. The parameter S can be used as an encoding parameter, and is typically selected based on image resolution, source frame rate and desired trade-off between complexity and quality. Generally, increasing the value of S results in a higher quality image, since a larger extent of the reference frame can be searched to find a best matching block. The choice of M and N in this matter is beneficial in that it ensures that the phase correlation block is large enough to fully cover the entire search area for a macroblock of size 16x16 located at the center of the phase correlation block. (A macroblock of size 16x16 is the largest block size that is used for prediction in existing H.26x and MPEG-1/2/4 video compression standards.) If the maximum allowable search range in the horizontal direction is S_h, then

$$M > 2S_h + 16 \tag{1}$$

[0025] Similarly

$$N > 2S_v + 16 \tag{2}$$

[0026] where S_v is the maximum search range in the vertical direction.

[0027] As an example, if $S_v = S_h = 32$, then the phase correlation block will be of size 128x128.

[0028] As an optional step, the motion estimator 122 applies 202 a windowing filter to each phase correlation block. The windowing filter process is further described below.

[0029] Next, the motion estimator 122 iterates over the co-located phase correlation blocks of the reference frame R_{ij} and the current frame P_{ij}, where *i* is the number of columns of phase correlation blocks, and *j* is the number of rows of phase correlation blocks, and obtains 204 phase correlation values for each pair of phase correlation blocks. The phase correlation analysis is as follows.

[0030] Let $f_2(x, y)$ be a function that represents the phase correlation block in current frame P for which motion estimation is to be done and $f_1(x, y)$ be the function that represents the reference frame R. Assuming translation motion, $f_2(x, y)$ can be expressed as

$$f_2(x, y) = f_1(x - x_0, y - y_0) \tag{3}$$

[0031] If $F_1(u,v)$ and $F_2(u,v)$ are the Fourier Transforms (FT) of $f_1(x,y)$ and $f_2(x,y)$ respectively, a shift in the spatial domain is reflected as a phase shift in the frequency domain and can be written as:

$$F_2(u,v) = F_1(u,v)e^{(-i(ux_0+vy_o)} \tag{4}$$

[0032] The product of the two FTs gives a measure of the phase correlation between the two frames. The normalized cross-power spectrum can be written as

$$\frac{F_1(u,v)F_2^*(u,v)}{|F_1(u,v)F_2^*(u,v)|} = e^{(i(u\,x_0 + vy_0))} \tag{5}$$

[0033] The Inverse Fourier transform (IFT) of the normalized cross-power spectrum is then used to extract the motion vector (x_0,y_0) from (5) since the resulting correlation surface has a unit impulse response centered at (x_0,y_0).

$$F^{-1}\left[\frac{F_1(u,v)F_2^*(u,v)}{|F_1(u,v)F_2^*(u,v)|}\right] = \delta(x_0, y_0) \tag{6}$$

[0034] Thus, the motion estimator 122 performs an FFT on each of the phase correlation blocks P_{ij} and R_{ij}, determines their normalized cross product, and takes the IFT of this value to determine a phase correlation surface. The motion estimator 122 then identifies 206 the location of the *K* highest peak(s) (i.e. those with the largest magnitude) in the phase correlation surface for the current phase correlation block. The locations of these peaks give K candidate motion vectors. Each motion vector is directed from the center of the phase correlation block to the (x,y) location of the peak.

[0035] If there is just a single, uniform motion and no object deformation, the IFT correlation plane has a single peak at (x_0, y_0), corresponding to the single motion vector. If there are multiple motions with several objects, or objects undergo deformation, the plane of IFT shows several peaks each representing a component motion, as illustrated in the example FIG. 5, where *K*=10; Table 1 of FIG. 5 shows the *K* candidate motion vectors for this example, with their peak values, and their (x,y) values for their associated motion vectors. Thus, by locating the different peaks in the correlation surface, the motion vectors corresponding to different component motions in the frames are identified.

[0036] The number *K* may be adjusted to trade-off encoding speed and image quality. In different embodiments, *K* can be selected as 1) a fixed value (e.g. 10 or 16); 2) as a function of the size of the block to be predicted (the larger the size, the smaller the value of K); 3) based on the distribution of the peaks in the correlation surface (with the greater the variance in the values of the peaks, the smaller the value of *K*). This last case generally entails selecting a smaller number of candidates when there are a few very dominant peaks, and selecting more candidates if there are no peaks that are clearly dominant. Each of these alternatives can be used independently, or in any combination, as a weighted function.

[0037] In another embodiment, the motion estimator 122 performs motion estimation at sub-pixel resolution using the phase correlation approach, for half-pixel accuracy, for instance. In this embodiment, the phase correlation surface given in Eq. (6) is interpolated by a factor of two in each direction.

[0038] Referring again to FIG. 2, the motion estimator 122 uses the *K* candidate motion vectors for each predicted phase correlation P_{ij} to identify 208 the best motion vector for each macroblock in the phase correlation block P_{ij}. For each phase correlation block P_{ij}, the motion estimator 122 searches the *K* candidate motion vectors for each block (a block can be either a macroblock or sub-block) within the phase correlation block to select 208 the best motion vector. More particularly, for each block in the current phase correlation block P_{ij}, the motion estimator 122 applies each of the *K* candidate motion vectors to the current block, offsetting the block's location onto a block of the reference frame. The motion estimator 122 determines a distortion measure (e.g., sum of absolute difference, SAD) between the current block of the current frame P relative to the predictor block of reference frame R. The candidate motion vector that minimizes distortion is chosen to be the best candidate to represent the motion between the two blocks. When the search is completed, the motion estimator 122 outputs the set 212 of motion vectors for the input frame to the motion compensation stage 124.

[0039] An as optional embodiment, once the motion estimator 122 identifies the best candidate out of the *K* candidates for a current block, it then performs a fractional motion search 210 around the selected candidate for sub-pixel accuracy. In one embodiment, the motion estimator 122 searches the eight neighboring half-pixel motion vectors. In another embodiment, a secondary subpixel search is done at the eight quarter-pixel motion vectors surrounding the best candidate at the end of the half-pixel search. Thus, even with both subpixel searches, the total number of motion vectors searched is limited to K + 16, regardless of the size of the search area.

[0040] As another optional embodiment, a further reduction in the number of candidate vector evaluations can be obtained by early termination 214 of the motion vector evaluation if the distortion measure is less than a predefined or adaptive threshold value. If this condition is encountered, no more candidates need to be tested for further integer pel search and for half/quarter pixel search. Conditions for early termination may include 1) reaching a pre-determined, fixed distortion measure; 2) reaching a distortion measure threshold that is derived from the distortion measure of the previously encoded neighboring blocks. For instance, the threshold could be the minimum of the distortion measures of the three neighboring blocks or blocks that lie directly above, directly to the left and above-right (northeast) of the block to be encoded.

[0041] This phase correlation technique directly measures the magnitude and direction of motion vectors rather than estimating them using block-matching searches. As such, the complexity of performing phase correlation is essentially independent of the motion search area. The number of candidate motion vectors that need to be tested does not vary with the change in the motion search window size. Hence there is no addition in complexity by increasing the search window, which is of primary importance for high quality encoding of higher resolution sequences.

[0042] Another benefit of the phase correlation method is that it is also insensitive to variations in intensity due to the normalization step (Eq. 5) leading to a whitening of signals. This implies that phase correlation gives a true measure of the movement of objects between two frames. Due to this property it is possible that the motion vectors for neighboring blocks might be more uniform than would be the case for FME algorithms based on the shape of the search pattern. This property makes it more suitable for rate distortion optimization.

[0043] As indicated above, in one embodiment, the phase correlation blocks are non-overlapping. Phase correlation depends on identifying the relative motion between objects so naturally its performance would be significantly better if the entire object is encompassed within the phase correlation block in both frames. Since this cannot be guaranteed in video sequences, in one embodiment, the phase correlation blocks are allowed to have a minimum overlap value of μ_h and μ_v pixels in the horizontal and vertical directions, respectively, $\mu_h \geq S_h$ and $\mu_v \geq S_v$. This ensures that if any object moves from one phase correlation block to another, then the motion vectors from at least one of the overlapping phase correlation blocks would provide a good match. In this embodiment, for those blocks that partly lie in more than one phase correlation block, the motion estimator 122 tests 208 the *K* candidate motion vectors from all the phase correlation blocks in which the block is partly located in order to select the best motion vector.

[0044] The μ_h and μ_v pixel overlap minimums also ensures that objects that are up to S_hxS_v pixels large are guaranteed to completely fall into one or another phase correlation block. A much larger overlap would result in there being more phase correlation blocks, thereby resulting in a higher complexity. The amount of overlap can thus be viewed as yet another parameter for trading off complexity for compression efficiency.

[0045] In one embodiment, the amount of overlap is selected in such a manner that the phase correlation blocks are evenly distributed across the frame. Referring to FIG. 4, given a frame of size *W*x*H* (where the frame could be either the actual frame or the extended frame), and given phase correlation blocks 300 of size *M*x*N*, and given an overlap of □$_h$ in the horizontal direction between two adjacent phase correlation blocks, results in the following relationship:

$$W = M + (i-1)(M - \mu) \tag{7}$$

where *i* is an integer number of phase correlation blocks in one row. A minimum □$_h$ pixel overlap in the horizontal dimension implies that $\mu_h \geq S_h$. Replacing this value in Eq. (7) and solving for *i* yields:

$$i \geq \frac{(W-M)}{M-S_h} + 1 \tag{8}$$

[0046] This implies that *i* is the smallest integer that satisfies the inequality above. Once *i* is obtained, the amount of overlap □$_h$ can be deduced from Eq. (1) by solving for □$_h$:

$$\mu_h = M - \frac{W-M}{i-1} \tag{9}$$

[0047] Using the same process *j*, the number of phase correlation blocks in one column can be derived, and therefore □*v*, the amount of overlap in the vertical direction:

$$j \geq \frac{(H-N)}{N-S_v} + 1 \tag{10}$$

$$\mu_v = N - \frac{H-N}{j-1} \tag{11}$$

[0048] As indicated above, an optional process in the phase correlation analysis is the application 202 of a windowing filter to the phase correlation blocks. The rectangular shape of the phase correlation blocks creates a sharp signal discontinuity at the edges of these blocks. This discontinuity produces extra high-frequency components in the FFT spectrum domain, potentially leading to false peaks in the phase correlation surface. In one embodiment, these spurious peaks are reduced by smoothing out the pixel values at the edges of the phase correlation blocks using a windowing filter. This may be done, for example, by applying a suitable 2D windowing function to scale down the pixel values at the edges of each phase correlation block. In one embodiment a modified, 2-D version of the 1D extended cosine bell window is used. FIG. 6 illustrates a 1D modified extended cosine bell window. Preferably, the transition regions of the cosine bell are set to be equal to 1/16 of phase correlation block size; this leaves the middle 7/8ths of the phase correlation block scaled at 100%. Pixels within each transition region are scaled by the corresponding curve value. For example, for a phase correlation block where M=N=128, the transition region T equals 128/16 = 8 pixels. The 2D cosine bell applies the 1D version in both the horizontal and vertical directions, with the resulting scale value being the product of the individual curve scale values. Assuming that windowing has been used amount of the overlap from Eqs. (9) and (11) may be increased by the amount of T. Equations (12) and (13) define the 1-D and 2-D versions of the windowing filter for one embodiment:

[0049] Modified 1D Extended Cosine Bell Window *W(n)* of Phase Correlation Block of size N:

$$W(n)=\begin{cases}\frac{1}{2}\left[1-\cos\left(\frac{16*n*\Pi}{N}\right)\right]\ldots for\left(\frac{N}{16}\leq n\ldots or\ldots n\geq\frac{15*N}{16}\right)\\ 1\ldots otherwise\end{cases}\qquad(12)$$

[0050] Modified 2D Extended Cosine Bell Window *W(m,n)* of Phase Correlation Block of size MxN:

$$W(m,n)=\begin{cases}\frac{1}{2}\left[1-\cos\left(\frac{16*m*\Pi}{M}\right)\right]*\frac{1}{2}\left[1-\cos\left(\frac{16*n*\Pi}{N}\right)\right]\\ \quad for\left(\frac{M}{16}\leq m\ldots or\ldots m\geq\frac{15*M}{16}\right)\ and\ \left(\frac{N}{16}\leq n\ldots or\ldots n\geq\frac{15*N}{16}\right)\\ 1\ldots otherwise\end{cases}\qquad(13)$$

[0051] In summary then, the present invention provides a way of determining the best motion vector for predicting each block in current frame by using the phase correlation information between blocks of the reference frame and the current frame. Rather then iteratively comparing blocks directly as in conventional methods, a motion estimation process in accordance with the present invention can first accurately select a set of potential motion vector candidates using phase correlation information. From this reduced set of candidates, a best motion vector can be readily identified.

[0052] The present invention has been described in particular detail with respect to one possible embodiment. Those of skill in the art will appreciate that the invention may be practiced in other embodiments. First, the particular naming of the components, capitalization of terms, the attributes, data structures, or any other programming or structural aspect is not mandatory or significant, and the mechanisms that implement the invention or its features may have different names, formats, or protocols. Further, the system may be implemented via a combination of hardware and software, as described, or entirely in hardware elements. Also, the particular division of functionality between the various system components described herein is merely exemplary, and not mandatory; functions performed by a single system component may instead be performed by multiple components, and functions performed by multiple components may instead be performed by a single component.

[0053] Some portions of above description present the features of the present invention in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. These operations, while described functionally or logically, are understood to be implemented by computer programs. Furthermore, it has also proven convenient at times, to refer to these arrangements of operations as modules or by functional names, without loss of generality.

[0054] Unless specifically stated otherwise as apparent from the above discussion, it is appreciated that throughout the description, discussions utilizing terms such as "calculating" or "determining" or "identifying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0055] Certain aspects of the present invention include process steps and instructions described herein in the form of an algorithm. It should be noted that the process steps and instructions of the present invention could be embodied in software, firmware or hardware, and when embodied in software, could be downloaded to reside on and be operated from different platforms used by real time network operating systems.

[0056] The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored on a computer readable medium that can be accessed by the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus. Those of skill in the art of integrated circuit design and video codecs appreciate that the invention can be readily fabricated in various types of integrated circuits based on the above functional and structural descriptions, including application specific integrated circuits (ASIC). In addition, the present invention may be incorporated into various types of video coding devices.

[0057] The algorithms and operations presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may also be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will be apparent to those of skill in the art, along with equivalent variations. In addition, the present invention is not described with reference to any particular programming language. It is appreciated that a variety of programming languages may be used to implement the teachings of the present invention as described herein, and any references to specific languages are provided for disclosure of enablement and best mode of the present invention.

[0058] Finally, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.